

FPGA Implementation of 16-bit Multipliers based upon Vedic Mathematic Approach

Zulhelmi

Jurusan Teknik Elektro, Universitas Syiah Kuala
Jl. Tgk. Syech Abdurrauf No. 7 Darussalam, Banda Aceh 23111
e-mail: zulhelmi@unsyiah.ac.id

Abstract—This paper proposes design and implementation of a 16-bit multiplier based upon Vedic mathematic approach, where the design has been targeted to the Xilinx Field Programmable Gate Arrays (FPGAs) board, device XC5VLX30. The approach is different from a number of approaches that have been used to realize multipliers. It has been reported that previous algorithms such as Booth, Modified Booth, and Carry Save Multipliers only suitable for improving speed or decreasing area utilization; therefore, those algorithms are not appropriate for designing multipliers that are used for digital signal processing (DSP) applications. Moreover, they are not flexible to be implemented on FPGAs or on a single chip using application specific integration circuits (ASICs). Vedic approach, on the other hand, can be used to design multipliers with optimum speed and less area utilization. In addition, it is reliable to be implemented on FPGAs or on a single chip. Behavioral and post-route simulation results prove that the proposed multiplier shows better performance in terms of speed compared to the other reported multipliers when being implemented on the FPGA. In terms of area utilization, better results are also obtained.

Keywords: *multiplier, FPGA, Vedic mathematics, DSP, ASICs*

Abstrak—Naskah ini mengajukan rancangan dan implementasi 16-bit multiplier berdasarkan pendekatan matematika Vedic, dimana rancangannya ditargetkan untuk diimplementasikan pada Xilinx Field Programmable Gate Arrays (FPGAs) board, device XC5VLX30. Pendekatan ini berbeda dibandingkan dengan pendekatan yang pernah digunakan untuk merealisasikan multipliers. Pendekatan-pendekatan sebelumnya seperti Booth, Modified Booth, dan Carry Save Multipliers yang telah dipublikasikan hanya cocok untuk meningkatkan kecepatan atau mengurangi penggunaan area; oleh karena itu, pendekatan-pendekatan tersebut tidak cocok untuk desain multipliers yang digunakan pada aplikasi pemrosesan sinyal digital (DSP). Lebih lanjut, pendekatan-pendekatan tersebut juga tidak fleksibel untuk diimplementasikan pada FPGAs atau pada satu single chip dengan menggunakan aplikasi khusus rangkaian terpadu (ASICs). Sebaliknya, pendekatan Vedic dapat digunakan untuk merancang multipliers dengan kecepatan yang optimum dan penggunaan area yang lebih rendah. Hasil-hasil simulasi Behavioral dan post-route membuktikan bahwa multiplier yang diusulkan menunjukkan performa kecepatan perhitungan yang lebih baik dibandingkan dengan multipliers yang lain. Untuk penggunaan area saat implementasi, multiplier yang diusulkan juga mengindikasikan penggunaan area yang lebih rendah.

Kata kunci: *pengali, FPGA, matematika Vedic, DSP, ASICs*

I. INTRODUCTION

The requirement of high performance processors in digital signal processing (DSP) is increasing in new communication standards and high aggregation system. The largest silicon consumers of the DSP system are multipliers required by finite impulse response (FIR) filters and other DSP functions, so an efficient implementation of multipliers is the key for the cost-effective solution of these applications. In parallel with reducing the chip area required for multiplier implementation, the multiplier speed should be maintained or even increased during realization. These two dominant factors challenge researchers to do new works to find out the best multiplier that can be used in DSP applications. Multipliers are also important in matrix multiplications, which are applied, for instance, in 3D affine transformations [1].

A number of multipliers, demonstrating several advantages, have been reported in the last few decades. Goto et al. [2], for example, realized the regularly structured tree multiplier implemented using 0.8 μm CMOS process, focusing on layout density and multiplication time. Speed consideration is another example given by Lamberti et al. [3] who introduced a way of reducing computation time in two's complement multipliers with short bit width. Similar work also introduced by Dimitrov et al., who have developed efficient area multipliers based on multiple-radix representations [4]. The latter multipliers have been realized using 0.18 μm CMOS technology, and gave better area and power consumption compared to other multipliers. However, the technique is not suitable to build fast multipliers. Since the fabrication process is time consuming, an alternative approach as hardware realization is required to design hardware such multipliers.

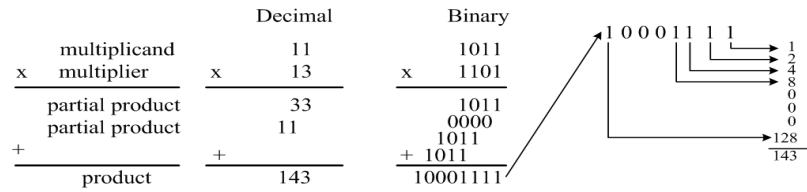


Figure 1. Basic multiplier concept

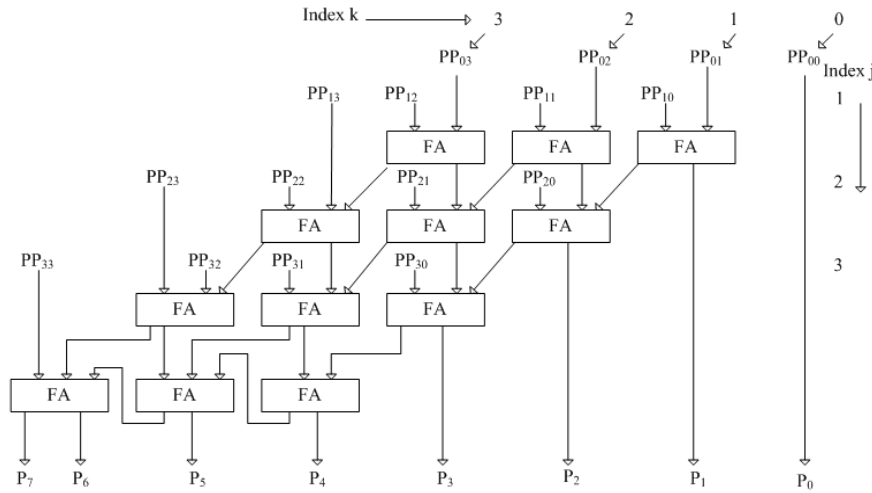


Figure 2. Block diagram for multiplication with CSA

Hence Field Programmable Logic Arrays (FPGAs) has been developed to solve the issues.

Prior to discovering FPGAs, multipliers have been designed and implemented on chips, as sub-systems of processors using Application Specific Integrated Circuit (ASIC). Even though the ASIC's implementation provides the best performance of realizing hardware, a number of issues is accounted as follows [5]:

- The Integrated circuit costs are rising aggressively
- ASIC complexity has lengthened development time
- R&D resources and headcount are decreasing
- Revenue losses for slow time-to-market are increasing
- Financial constraints in a poor economy are driving low-cost technologies

These trends make FPGAs a better alternative than ASICs for a larger number of even higher-volume applications than they have been historically used for. Further, it is well established that full custom ASICs are the most expensive to manufacture and design. They have the largest turn-around time. Thus they are preferred only when [6]:

1. There are no suitable existing cell libraries available that can be used for the entire design, i.e. the cells are not small or fast enough or consume too much power.
2. The ASIC technology is new or so specialized.

II. BACKGROUND

Since the requirement of multipliers is becoming important in DSP, a number of multiplication techniques

have been proposed to achieve the need of multipliers that have high speed and at the same time providing the low-power consumption as well as less area needed for its implementation.

A. Basic Multiplication

The traditional multiplication algorithm is basically illustrated in Figure 1. The figure indicates multiplication of two operands that are displayed in decimal and binary numbers. The first operand is called a multiplicand and the second one is called as a multiplier. The partial products are added using adders to get the final result. In binary number, the partial product is either zeros or multiplicand as can be seen from the figure.

As shown in the figure, there are two major steps to perform multiplication. First, it is having partial products from operands, and final step is adding partial products to attain final product. In terms of binary multiplication, one extra step, known as reduction method, is required to reduce a number of partial products.

B. Multiplication with Carry Save Adders

In multiplication, it is known that the partial products can be formed by an array of AND gates. Therefore, the multiplication can also be done by the scheme as shown in Figure 2. This algorithm is known as multiplication with carry save adder where the first full adder (FA), placed on the right, is kept the bit carry and then used in the second adder. Likewise first adder, the second adder is also kept the bit carry which will be used by third adder and so forth.

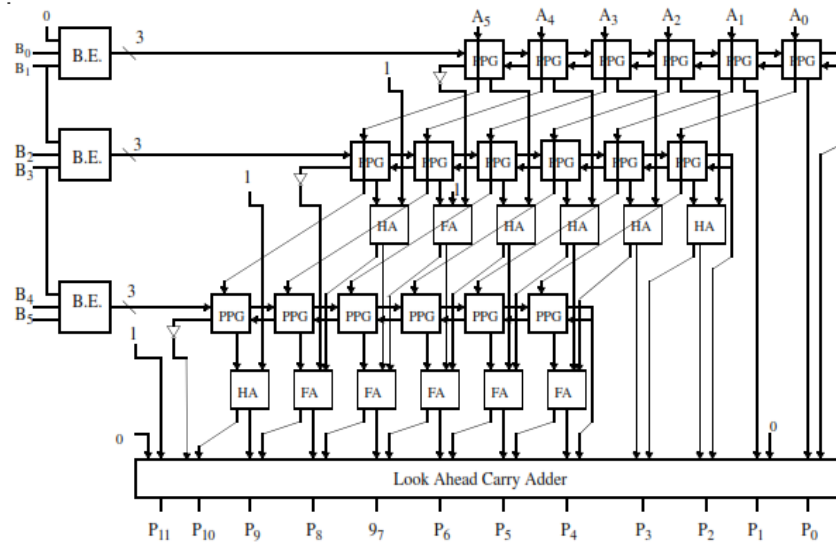


Figure 3. Block diagram for 6x6 MBE multipliers

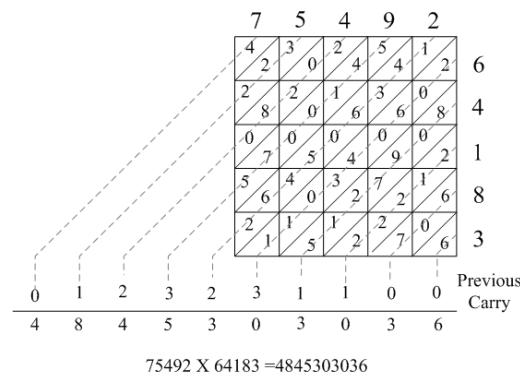


Figure 4. Vertical and crosswise formula in multiplication

C. Booth and Modified Booth Algorithms

Suggestions to improve the speed of a multiplier were offered by Booth and Mc Sorley in around 1960s. The algorithms are known as Booth Encoding (BE) and Modified Booth Encoding (MBE). Both techniques are meaningful to reduce Partial Products (PP) from the multiplication process. One example of MBE multipliers is depicted in Figure 3. In which, it requires three MBE circuits to encode six input bits of the multiplier. Partial product generation (PPG) yields three rows of PP, which are then compressed to two rows. Carry Look Ahead (CLA) adder has a task to perform final addition in order to get multiplication products as one row without redundant result.

The algorithms that have been discussed above have several drawbacks. For instance, basic multiplication algorithm is not suitable enough to realize multipliers since it yields weak performance multipliers. Multiplier with carry save adder, on the other hand, is fine when employed for applications that are not required high speed operation. Even though, booth and MBE multipliers offer well performance when applied to many digital applications, the complexity of hardware realization is a major issue. The multipliers are quite difficult to be implemented using

ASICs and FPGAs. Therefore, an alternative algorithm that gives good performance and eases to realize using ASICs and FPGAs is required.

III. METHOD

An alternative algorithm that is fit to DSP multipliers is multiplication based upon Vedic mathematics. This algorithm comes from the ancient Indian knowledge. Vedic mathematics contains several branches of knowledge. One of them is about multiplication concept. A formula that is very popular among the ancient Indian is "Urdhva Triyagbhyam" meaning vertical and crosswise [7]. This formula is clearly demonstrated in Figure 4 describing the multiplication process of two operands, 75492 by 64183.

The concept can be used to build well performance multipliers. Figure 5 indicates block diagrams for a 4-bit multiplier. It shows that the multiplier is built from four 2 by 2 multipliers (lower order multipliers). Each 2 by 2 multiplier is utilized to generate Partial Products (PP) labeled T_{00} - T_{03} , T_{10} - T_{13} , T_{20} - T_{23} , and T_{30} - T_{33} . The PP rows are resulted from multipliers M_0 - M_3 respectively. By arranging, according to its weighted values, the PP rows can be simplified as depicted in the figure. Final result is, then, attained by adding the PP rows.

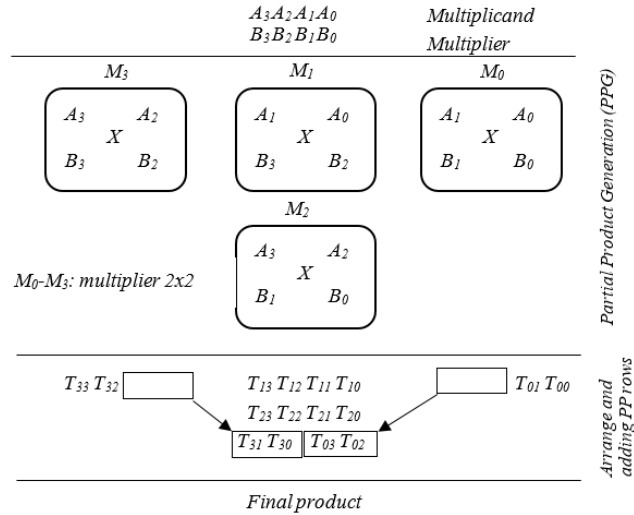


Figure 5. Block diagram for 4x4 multipliers based on Vedic mathematics concept

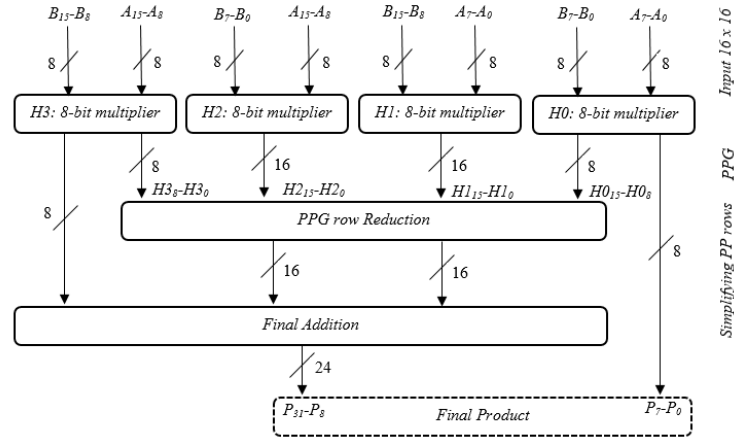


Figure 6. The proposed 16-bit multiplier

With this approach, higher order multipliers may be easily developed from lower order ones. Therefore, a 16-bit multiplier based on FPGA implementation is proposed. Since the 16-bit multiplier is the target implementation, lower order multipliers, in this case, 8-bit ones are required. Utilizing bottom up design, it is clearly understood that an 8-bit multiplier consists of 2-bit and 4-bit multipliers. Hence, the proposed multiplier can be drawn as indicated in Figure 6, where 2-bit and 4-bit multipliers are not depicted for simplicity.

Figure 6 shows that the proposed multiplier can be classified into four parts. Firstly, it is an input part, which has 16x16 inputs. The second part is generating three partial product rows that are performed by employing 8x8 multipliers (H_0, H_1, H_2 , and H_3). The partial products that consist of three rows is, then, simplified to two rows in the third part. Finally, final addition takes a task to produce final result. From block diagrams can be seen obviously that the 8-bit lower products are obtained directly after partial product generation (PPG). This is one advantage in the proposed design since the final addition part only requires to add more a 24-bit pair to obtain the final product. The speed of the proposed multiplier can be increased and area utilization may reduce significantly.

IV. RESULTS AND DISCUSSIONS

The proposed 16-bit multiplier and its corresponding blocks are described using structural Verilog-HDL and synthesized employing Xilinx Synthesis Tool (XST), WebPACK version 13.3. The implementation was targeted to Xilinx Virtex-5, device XC5VLX30. To check the functionality of the proposed multiplier, two types of simulations, behavioral and post-route, have been done using ISim that is a powerful tool in simulating digital designs. Figure 7 shows the behavioral simulation result that indicates the proposed multiplier has been designed properly. Similarly, the post-route simulation, depicted in Figure 8, shows that the design was succeeded implemented on a FPGA board. Close examination of Figure 8, redrawn in Figure 9, indicates the maximum delay of the multiplier about 9.15 ns. In other words, maximum allowed operating frequency is around 110 MHz if the FPGA board is operated together with other external devices.

From the implementation results as indicated in Table 1, it is found that the proposed 16-bit multiplier has a delay around 10 ns after implemented on the FPGA board, device Virtex-5 5v1x30ff324-3. It required 64 I/O pins and

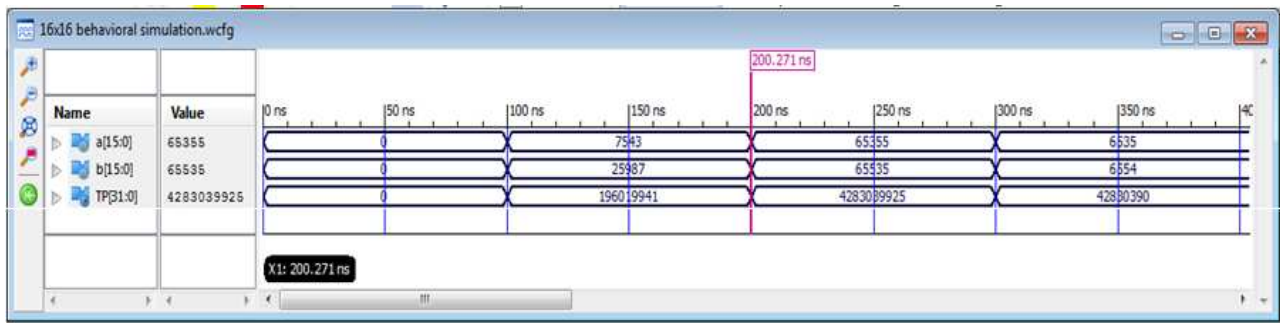


Figure 7. Behavioral simulation of the proposed multiplier

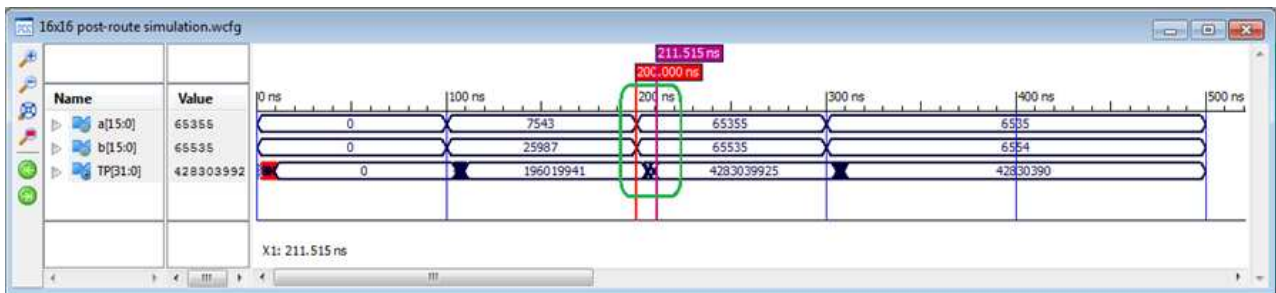


Figure 8. Post-route simulation of the proposed multiplier

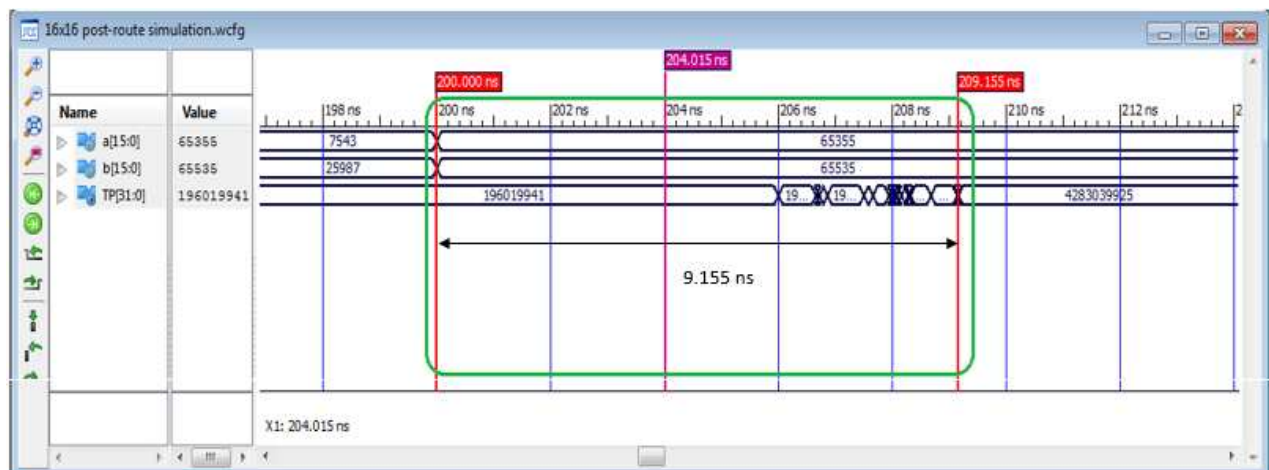


Figure 9. Close examination of the proposed multiplier

412 look up tables (LUTs) or equivalence to 105 slices.

Comparison to a previous work, the proposed multiplier provides, in terms of speed, better performance as illustrated in Figure 10. Multipliers that were reported by [8] have larger delay by more than two times for the case Basic, Carry Ripple, and Booth Signed multipliers compared with the proposed multiplier. Being compared to Carry Save Multiplier (CSM), the proposed multiplier is still superior with delay ratio about 0.64.

Table 1. Report summary of the proposed 16-bit multiplier

Selected Device	Type of Reports	No. of LUTs	No. of Slices	No. of IOs	Delay (ns)
Virtex-5	Synthesis	418	-	64	9.15
	Post and Route	412	105	64	10.67

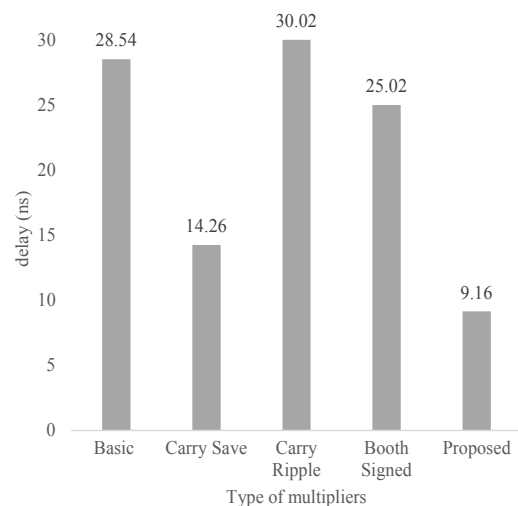


Figure 10. Delay comparison of 16-bit multipliers

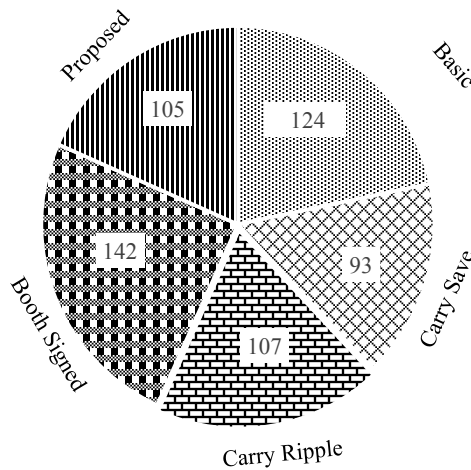


Figure 11. Area comparison (in slice) of 16-bit multipliers

In terms of area utilization, the proposed multiplier also shows better results. It can be seen clearly in Figure 11. The utilized area of the proposed multiplier consumes smaller number of slices compared to other three multipliers, and slightly higher than Carry Save Multiplier. In other word, the proposed multiplier has been implemented, in the most of case studied, with less area occupation.

V. CONCLUSION

The present work addresses a new approach for design and hardware realization of the 16-bit multiplier based on the Vedic mathematic concept. The approach employs lower order multipliers to develop a higher order one. In the case of building 16-bit multipliers, 2-bit, 4-bit, and 8-bit multipliers are utilized to generate partial products. In addition, partial product reduction and final adder

blocks are also used to yield final products. By optimizing each of lower order multipliers, the proposed multiplier shows better performance in terms of speed compared to the multipliers reported in literature. In terms of the device utilization also, in most of the cases studied, better results are obtained.

REFERENCE

- [1] F. Bensaali, A. Amira, I. S. Uzun, and A. Ahmedsaid, "An FPGA implementation of 3D affine transformations," in *Proceedings of IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, vol. 2, pp. 715-718, 2003.
- [2] G. Goto, T. Sato, M. Nakajima, and T. Sukemura, "A 54 X 54-b regularly structured tree multiplier," *IEEE Journal of Solid-State Circuits*, vol. 27, pp. 1229-1236, 1992.
- [3] F. Lamberti, N. Andrikos, E. Antelo, and P. Montuschi, "Reducing the computation time in (short bit-width) two's complement multipliers," *IEEE Transactions on Computers*, vol. 60, pp. 148-156, 2011.
- [4] Vassil S. Dimitrov, Kimmo U. Jarvinen, and Jithra Adikari, "Area-efficient multipliers based on multiple-radix representations," *IEEE Transactions on Computers*, vol. 60, pp. 189-201, 2011.
- [5] T. Erjavec, "Introducing the Xilinx targeted design platform: fulfilling the programmable imperative," White Paper, Xilinx, 2009.
- [6] M. J. S. Smith, *Application-specific integrated circuits*, Addison-Wesley, 1997.
- [7] H. D. Tiwari, G. Gankhuyag, K. Chan Mo, C. Yong Beom, "Multiplier design based on ancient Indian Vedic Mathematics," in *International SoC Design Conference (ISOCC)*, 2008.
- [8] S. Bhattacharjee, S. Sil, B. Basak, and A. Chakrabarti, "Evaluation of power efficient adder and multiplier circuits for FPGA based DSP applications," in *International Conference on Communication and Industrial Application (ICCIA)*, pp. 1-5, 2011.